

# Algoritmos y Estructuras de Datos II

## Tercer recuperatorio

Sábado 18 de diciembre de 2021

### Aclaraciones

- El parcial es individual en todas sus etapas, y a libro abierto.
- Se responderán consultas por Discord, sólo de interpretación del enunciado, a través del canal ‘Mesa de Docentes’ de la categoría PARCIALES.
- Cualquier aclaración sobre el enunciado se realizará a través del canal ‘anuncios-evaluaciones’.
- El parcial durará 4 horas, de 9 a 13hs, y la entrega de cada ejercicio se realizará a través del Campus hasta las 13:15hs, en forma estricta. Se desestimarán entregas fuera de tiempo, sin excepciones.
- El ejercicio se calificará con **Perfecto**, **Aprobado** (que puede ir con ‘–’ o ‘?’), **Regular**, o **Insuficiente**.
- Para aprobar la cursada se deben tener todos los ejercicios aprobados de todos los parciales.
- Los ejercicios con **R** o **I** se recuperarán por separado al final del cuatrimestre.
- Sólo se podrá tener un ejercicio con ‘A?’ en la cursada, si no, deberán recuperarse los que hagan falta.

### Ej. 1. Sorting

Cada año Papá Noel recibe cartas de los niños y las niñas en las cuales le piden una lista de regalos que les gustaría recibir. De forma de agilizar la entregas, el equipo de Papá Noel quiere ordenar los pedidos primero por zona horaria, de menor a mayor, y luego por cantidad de juguetes pedidos, de mayor a menor, de forma de ir entregando en horario, y liberando la mayor cantidad de regalos posible.

Las zonas horarias se representan como números de 1 a  $z$ , siendo 1 la zona en la que primero se hace Navidad y  $z$  la última. Además, se sabe que siempre se da la siguiente particularidad: del total de  $n$  niños y niñas que escriben en todo el mundo, por cada zona horaria hay como mucho mil niños y niñas que piden más de  $3n$  juguetes cada uno, mientras que todos los demás niños y niñas piden a lo sumo  $3n$  juguetes cada uno.

Papá Noel nos pidió que lo ayudemos con esta tarea, para lo que necesitamos un algoritmo para la función `ORDENENTREGAS`, que dada la cantidad de zonas horarias y un arreglo de tuplas de naturales, tales que cada una representa la zona horaria y la cantidad de juguetes que pidió cada niño o niña, devuelva el arreglo ordenado de la forma antes indicada.

$$\text{ORDENENTREGAS}(\text{in } z : \text{nat}, \text{in } \textit{pedidos} : \text{arreglo}(\text{tupla}(\text{nat}, \text{nat}))) \rightarrow \textit{res} : \text{arreglo}(\text{tupla}(\text{nat}, \text{nat}))$$

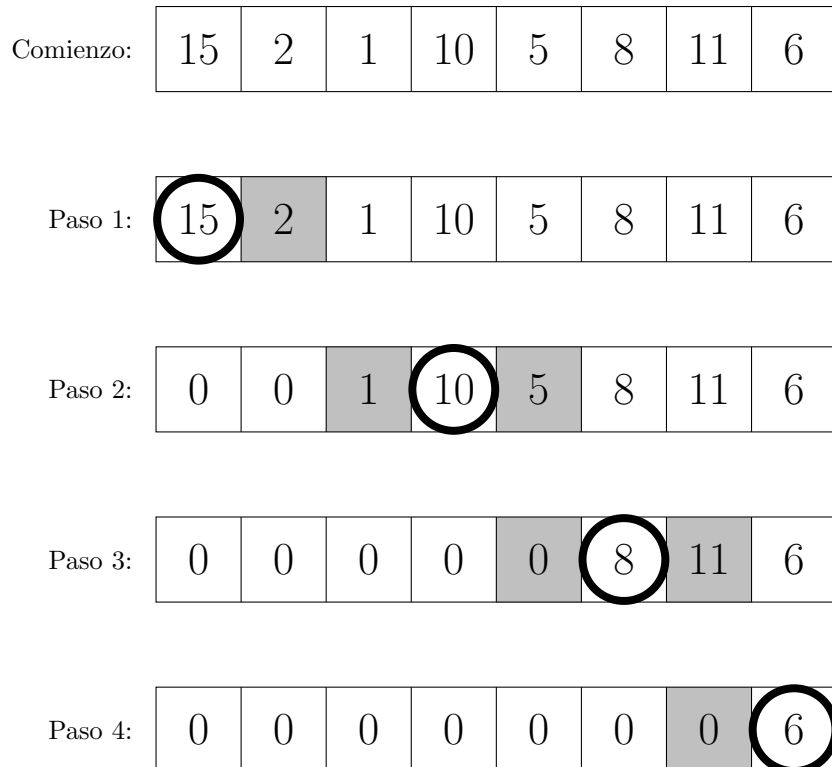
Así, se nos pide que este algoritmo tenga una cota de complejidad  $O(z + \text{long}(\textit{pedidos}))$ .

Escribir el pseudocódigo del algoritmo, justificar detalladamente que este resuelve el problema planteado y demostrar formalmente que cumple con la complejidad solicitada.

## Ej. 2. Divide and Conquer

Los elfos de Papá Noel tienen una fiesta de fin de año llena de juegos. Uno de estos juegos consiste en lo siguiente. Se tiene una fila de cajas que contienen alfajores. Cada caja indica el número de alfajores que contiene. El jugador puede elegir una caja (cualquiera, a gusto) y se lleva todos los alfajores que ella contiene, pero hay un pequeño truco: cuando elige una caja, una máquina vacía las cajas inmediatamente a la izquierda y a la derecha de la elegida. Luego, el jugador sigue eligiendo cajas hasta que no queden alfajores para tomar, que es el momento donde el juego termina y el elfo se retira con su premio. Posteriormente, se vuelven a llenar aleatoriamente las cajas para que pueda participar otro jugador.

La figura inferior ilustra esto, paso por paso. Cada celda representa una caja y el número de alfajores que contiene. En cada paso, la caja elegida se marcó con un círculo y se sombrearon las cajas que se vacían. Luego de 4 pasos, el juego termina y jugador tomó  $15 + 10 + 8 + 6 = 39$  alfajores.



Uno de los elfos que está haciendo fila para jugar quiere ganar a toda costa la mayor cantidad de alfajores que sea posible. Nos pidió que le facilitemos un algoritmo para la función `MAXALFAJORES`, que dado un arreglo con las cantidades de alfajores en las cajas devuelva la mayor cantidad de alfajores que se pueden conseguir en el juego:

$\text{MAXALFAJORES}(\text{in } \text{cajas} : \text{arreglo}(\text{nat})) \rightarrow \text{res} : \text{nat}$

### Ejemplos:

- $\text{MAXALFAJORES}([1, 3, 4, 3]) = 6$
- $\text{MAXALFAJORES}([15, 2, 1, 10, 5, 8, 11, 6]) = 39$

Nos pide además que este algoritmo utilice la técnica de Divide&Conquer y que tenga complejidad temporal  $O(n)$ , siendo  $n$  la cantidad de cajas.

Escribir el pseudocódigo del algoritmo, justificar detalladamente que este resuelve el problema planteado y demostrar formalmente que cumple con la complejidad solicitada. Para esto último, se deberá utilizar alguno de los métodos vistos de acuerdo a las características del problema que se está resolviendo.

**Nota:** pueden suponer que  $n$ , la cantidad de cajas, es una potencia de dos.

**Pista:** quizás sea necesario calcular varias cosas auxiliares al mismo tiempo.